CLAIM AMENDMENTS

Claim Amendment Summary

Claims pending

- Before this Amendment: Claims 1, 4-17 and 19-34.
- After this Amendment: Claims 1, 4-17, and 19-25.

Non-Elected, Canceled, or Withdrawn claims: Claims 2, 3, 18 and 26-34.

Amended claims: Claims 1, 4-7, 17 and 23.

New claims: None.

Claims:

1. (Currently Amended) A programming interface embodied on one or more computer readable media having computer-executable instructions for performing steps comprising:

generating graphical objects using a first group of services;

formatting content using a second group of services;

creating components of the graphical objects using a third group of services;

binding elements to data sources, data source classes, and data specific implementations of data collections using a fourth group of services, wherein the fourth group of services further handle exceptions in data entry;

using a common markup language to map classes and properties specified in the markup language to an instantiated tree of objects across the first group of services, the second group of services [[and]] the third group of services, and the fourth group of services; and

integrating the first group of services, the second group of ~~service~~ services, [[and]] the third group of services, and the fourth group of services using a consistent programming model and consistent services across the three service groups.

2. (Canceled)

3. (Canceled)

4. (Currently Amended) A programming interface as recited in claim 1, wherein the first group of services, the second group of services, [[and]] the third group of services, and the fourth group of services share a common event system.

5. (Currently Amended) A programming interface as recited in claim 1, wherein the first group of services, the second group of services, [[and]] the third group of services, and the fourth group of services share a common property definition system.

6. (Currently Amended) A programming interface as recited in claim 1, wherein the first group of services, the second group of services, [[and]] the third group of services, and the fourth group of services share a common input paradigm.

7. (Currently Amended) A programming interface as recited in claim 1, wherein the first group of services, the second group of services, [[and]] the third group of services, and the fourth group of services share a common system for nesting elements associated with a particular group of services within elements associated with another group of services.

8. (Original) A programming interface as recited in claim 1, wherein the first group of services includes a service that determines an appearance of the graphical objects.

9. (Original) A programming interface as recited in claim 1, wherein the first group of services includes a service that determines a behavior of the graphical objects.

10. (Original) A programming interface as recited in claim 1, wherein the first group of services includes a service that determines an arrangement of the graphical objects.

11. (Original) A programming interface as recited in claim 1, wherein the first group of services includes a plurality of nested elements that define the graphical objects.

12. (Original) A programming interface as recited in claim 1, wherein the graphical objects are comprised of one or more elements defined by vector graphics.

13. (Original) A programming interface as recited in claim 1, wherein the first group of services can define window properties in a markup language without launching a new window.

14. (Original) A programming interface as recited in claim 1, wherein the first group of services generate a user interface containing a plurality of graphical objects.

15. (Original) A programming interface as recited in claim 1, wherein the second group of services arrange the graphical objects.

16. (Original) A software architecture comprising the programming interface as recited in claim 1.

17. (Currently Amended) An application program interface embodied on one or more computer readable media having computer-executable instructions for performing steps comprising:

generating graphical objects using a first group of services;

formatting content using a second group of services;

creating components of the graphical objects using a third group of services;

binding elements to data sources, data source classes, and data specific implementations of data collections using a fourth group of services, wherein the fourth group of services further handle exceptions in data entry;

wherein the first group of services, the second group of services, [[and]] the third group of services, and the fourth group of services are integrated via:

sharing a common programming model; and

using a common markup language across the three services to map classes and properties specified in the markup language to an instantiated tree of objects.

18. (Canceled)

19. (Original) An application program interface as recited in claim 17, wherein the third group of services includes services to generate geometric shapes.

20. (Original) An application program interface as recited in claim 17, wherein the second group of services includes arranging a plurality of data elements.

21. (Original) An application program interface as recited in claim 17, wherein the first group of services includes:

a service that determines an appearance of a graphical object; and

a service that determines a behavior of the graphical object.

22. (Original) An application program interface as recited in claim 17, wherein the first group of services includes a service that defines window properties in a markup language without launching a new window.

23. (Currently Amended) A computer system including one or more microprocessors and one or more software programs, the one or more software programs utilizing a programming interface to request services from an operating system, the programming interface including separate commands to request services consisting of the following groups of services:

a first group of services for generating graphical objects; [[and]]

a second group of services for creating components of the graphical objects; and

a third group of services that bind elements to data sources, data source classes, and data specific implementations of data collections, wherein the third group of services further handle exceptions in data entry;

the first group of services, [[and]] the second group of services, and the third group of services are integrated by sharing a common programming model, consistent services and using a common markup language to map classes and properties specified in the markup language to an instantiated tree of objects across the first, [[and]] second, and third group of services.

24. (Original) A computer system as recited in claim 23, wherein the first group of services includes:

a service for defining an appearance of the graphical objects; and

a service for defining an arrangement of the graphical objects.

25. (Original) A computer system as recited in claim 23, wherein the second group of services includes services to generate a plurality of geometric shapes.

26 - 34. (Canceled)